AD^3-GLaM: A Cooperative Distributed QoE-based Approach for SVC Video Streaming over Wireless Mesh Networks

Pham Tran Anh Quang[a,*], Kamal Deep Singh[b], Juan Antonio Rodríguez-Aguilar[c], Gauthier Picard[d], Kandaraj Piamrat[e], Jesús Cerquides[c], César Viho[a]

[a] *IRISA, University of Rennes 1, Campus de Beaulieu, 35042 Rennes, France*
[b] *Laboratoire Hubert Curien, Université de Saint-Etienne, Jean Monnet, F-42000, Saint-Etienne, France*
[c] *IIIA-CSIC, Campus de la UAB, Bellaterra, Barcelona*
[d] *Laboratoire Hubert Curien, University of Lyon, MINES Saint-Etienne, F-42023, Saint Etienne, France*
[e] *LS2N, University of Nantes, Nantes, France*

Abstract

We study the routing problem of scalable video coding video streaming over wireless mesh networks. In contrast to most of the conventional routing algorithms, our proposal focuses on optimizing users' satisfaction. The mean opinion score –an indicator of quality of experience (QoE) in video streaming– is utilized to assess the quality of routes in wireless mesh networks. The objective is to optimize the overall user experience in the network. Conventional routing approaches do not consider QoE and are not optimal with respect to user experience. Moreover, some centralised approaches are not scalable and require significant computational resources. The latter disadvantage can be overcome using distributed approaches. This paper presents a QoE-based cooperative distributed routing approach. Among distributed cooperative optimization schemes, AD^3 is highlighted as one of the most efficient because of its fast convergence. The contributions of this paper are as follows: we encode the original problem into a factor graph and optimize the number of exchanged messages; we propose a partially distributed routing scheme based on OLSR and AD^3; and we propose a distributed decoding algorithm in order to find a feasible solution. Our thorough simulation results confirm the advantages of the proposed scheme.

*Corresponding author at: IRISA, University of Rennes 1, Campus de Beaulieu, 35042 Rennes, France.

Email address: quang.pham-tran-anh@irisa.fr (Pham Tran Anh Quang)

1. Introduction

The recent rapid growth of wireless networks has led to the development of various wireless applications that are widely used in our modern lives, in various domains such as military, commerce, entertainment, etc. Moreover, the number of wireless Internet users has been drastically increasing [1]. Furthermore, wireless local area network (WLAN) communication interfaces can be found on most devices nowadays. Consequently, wireless devices can connect to one another and form wireless mesh networks (WMNs). Recent works in the literature have proposed interesting applications which confirm the benefits of adopting WMNs [2, 3, 4, 5]. The abundance of wireless links can be exploited in several scenarios. For example, wireless mesh networking can be utilized to maintain connections in disaster recovery scenarios, when conventional infrastructure networks are unavailable.

Video streaming is one of the most popular services on the Internet and its traffic accounts for 70% to 82% of all Internet traffic [6]. Scalable video coding (SVC) is an extension of the H.264/MPEG-4 AVC video compression standard. It enhances flexibility of video streaming over inherently unstable wireless networks [7]. The major advantage of SVC is its capability of decoding a stream with partially received data [8]. The more layers a client can receive, the better the video quality. The video quality measurement can be obtained either using objective or subjective approaches. The subjective approaches are based on evaluations done by real humans. Thus, they are more correlated to users' experience than objective approaches, which are based on network-oriented metrics. Regarding subjective quality-assessment methods, the International Telecommunication Union (ITU) recommends the Mean Opinion Score (MOS) metric [9]. MOS can be divided into five levels corresponding to the users' perception: 5 (Excellent), 4 (Good), 3 (Fair), 2 (Poor), 1 (Bad). MOS is a good measure of quality, but it requires a lot of resources and cannot be obtained automatically. A hybrid Quality of Experience (QoE) evaluation method, named Pseudo-Subjective Quality Assessment (PSQA), was proposed in [10] to estimate users' experience in real-time. The most recent version of PSQA for SVC was introduced in [11].

In WMNs, traffic flows may have to traverse through multiple relaying nodes until reaching the destination. That can deteriorate the performance as well as the quality of some applications, especially video streaming ones. Determining the end-to-end path that can enhance user experience is the most important task in WMNs. Because of the aforementioned advantages of SVC and QoE metrics and the importance of routing in WMNs, we study QoE-based routing problem for SVC video streaming over WMNs in this paper. A use-case of this scenario is video streaming in rural areas where the cellular networks may not be available or may have insufficient coverage. Thanks to the wide-spread of wired networks, the dwellings may be equipped with gateways (GWs) connecting to the Internet through high speed connections. Consequently, WMNs with multiple GWs are considered in this study.

The existing routing schemes can be categorized into two groups: central-

ized [12, 13, 14, 15, 16] and distributed [17, 18, 19]. The centralized routing algorithms contain a centralized entity. The centralized controller is able to characterize the whole network and determine the global optimal or sufficiently sub-optimal routing solutions. In contrast, the distributed routing algorithms allow nodes to find out the local optimal solutions based on their knowledge. Centralized schemes can provide high quality routing solutions, however, they also have some disadvantages, such as high requirements in terms of computational resources, high calculation time, etc. Conventional distributed schemes can deal with large-scale network, but its routing configuration is far from the optimal solution.

Alternating Directions Dual Decomposition (AD^3) [20], an algorithm proposed in the realm of the machine learning literature, has been empirically shown to outperform state-of-the-art message-passing algorithms on a wide variety of large-scale problems. Furthermore, existing machine learning approaches that have been particularly designed for routing come with high cost in terms of time [21], e.g. thousands of seconds, which is far above our requirements. Nonetheless, the applicability of AD^3 to distributedly solve an optimization problem poses non-trivial challenges: (1) an encoding of the optimization problem as a so-called factor graph, a graph-based structure, that guarantees the efficient computation of messages by AD^3; (2) the operation of AD^3 in a distributed manner; and (3) the *decoding* of the solution (using LP relaxation of the optimisation problem) obtained by AD^3 into a feasible solution.

Here we address the above challenges through the following contributions:

- We cast the routing problem in WMNs as an optimization problem
- We provide an encoding of the optimization problem as a factor graph. The encoding employs AD^3 computationally-efficient factors in order to guarantee efficient computation.
- We formulate and solve a factor and variable assignment problem in order to optimize the number of messages exchanged by GWs.
- We propose a scheme that is based on combining the well-known OLSR, to gather information about the network, with AD^3, to solve the optimization problem in a distributed manner.
- We design a distributed decoding algorithm to convert the solution output by AD^3 into a feasible solution.

The rest of this paper is organized as follows. Section 2 provides related works. Section 3 outlines the whole proposed scheme. Section 4 provides mathematical descriptions of multi-channel WMNs under time constraints. In Section 5, we demonstrate how to convert the optimization problem into a factor graph and solve it with AD^3. A joint variable and factor assignment problem is also studied in this section. Subsequently, the output is decoded by the GLaM algorithm, which is presented in Section 6. Simulation results are presented and discussed in Section 7. Finally, Section 8 concludes the paper.

2. Related Work

In WMNs, routing is one of the important elements that impacts the overall performance. Therefore, several routing algorithms have been proposed and are discussed in this section.

2.1. Centralized vs. Distributed Routing Algorithms

We categorize existing routing algorithms into two major types: centralized and distributed. With distributed algorithms, each entity takes decisions independently, based on the local information available. Ad-hoc On-Demand Distant Vector (AODV) [22] and Optimized Link State Routing Protocol (OLSR) [23] are the most well-known protocols in this type. Both of them determine the path to a destination based on the number of hops. The main difference is that OLSR maintains a routing table at every node while AODV creates and maintains the routes when they are needed. OLSR is more efficient in high density networks [24]. A multipath extension of OLSR was presented in [25]. Multiple end-to-end paths are determined explicitly at the source by Multipath Dijkstra Algorithm. Though the routes in MPOLSR are not computed distributedly, the distributed selection of Multipoint relays (MPRs) puts MPOLSR into decentralized group. Another variant of OLSR, named cross-layer QoS-aware routing protocol on OLSR (CLQ-OLSR), has been introduced in [26]. Two sets of routing mechanisms were implemented, physical modified OLSR protocol (M-OLSR) and logical routing, by constructing multi-layer virtual logical mapping over physical topology. Physical M-OLSR protocol is responsible for routing table construction and bandwidth estimation on best-effort interface, while logical routing on real-time interfaces computes the optimized logical path using topology and bandwidth information. Every node in CLQ-OLSR estimates the available bandwidth on each associated channel. Each node disseminates the information of topology and available bandwidth to other nodes through HELLO and TC messages. The optimized logical path could be computed using the topology and bandwidth information. CLQ-OLSR outperforms OLSR [23] and multichannel-OLSR [27] in terms of average packet delivery rate, delay, and jitter. Above routing algorithms consider some QoS parameters, however, they did not take video specific parameters into account. Consequently, the routing configurations of these algorithms do not provide optimal performance for video streaming services. Whereas, such video specific parameters are considered in our work. Furthermore, some routing protocols are unable to provide the optimal solutions as shown in Section 7.

In [28], the algorithm assigns different paths and different transport layer protocols to different types of frames. The I-frames and inter-frames are conveyed by TCP and UDP respectively. The paths for I-frames are determined by adopting TCP-ETX routing metric while inter-frames are transported through the shortest path. Although this scheme is able to enhance the reliability, it does not consider user experience. A novel routing algorithm, called Quality of Experience (QoE) Q-learning based Adaptive Routing (QQAR), was presented in [29]. QQAR takes experience of users into account. QoE measurement is

integrated into routing paradigm in order to enable adaptive and evolutionary capabilities of the system. QQAR utilizes PSQA tool for QoE evaluation and Q-Learning algorithm for determining the paths. The simulations confirmed that QQRA outperforms other traditional approaches in terms of MOS. The algorithm was not designed for wireless networks and SVC videos. However, this routing algorithm was designed for wired networks.

With centralized routing algorithms, a central entity is responsible for determining all routes in the networks. A joint conflict-free routing and scheduling for real-time traffic in WMNs was studied in [13]. The authors proposed a fast and accurate algorithm called Delay-Aware Routing and Scheduling (DARS). However, it does not consider multi-path routing which can split a stream into different paths to enhance the streaming performance. In [12], an optimization architecture for joint multi-path and scheduling problem is proposed for WMNs. An optimal route is selected by considering link interference. Nevertheless, all of these aforementioned studies do not take user's perception into account, even though perceptual quality is very important for video streaming.

In [14], the authors adopted constrained shortest path first to determine the end-to-end paths that satisfy given requirements of throughput, loss rate, and jitters for video streaming over Software-defined networks. It is worth noting that the algorithm takes quality of experience (QoE) into account. The values of QoE metric are monitored and a new path is determined when there is degradation of QoE. The scheme, nevertheless, was designed for wired networks. In previous work [15, 16], we proposed the centralized algorithm of QoE-based routing in WMNs. However, the major disadvantages of centralized algorithms are high computational complexity and overhead. Therefore, in this paper, we study a distributed cooperative algorithm to overcome these shortcomings.

In WMNs, the availability of multiple GWs is possible. An intelligent gateway selection can improve the capacity of networks [30, 31]. A Gateway-aware Routing Metric (GARM), which can effectively select the best gateway for each node, was presented in [30]. A smart gateway selection is able to enhance the effective capacity of the network. In [31], two novel cross-layer control algorithms called Cross-Layer Control algorithm with Dynamic Gateway Selection (CLC_DGS) and Cross-Layer Control algorithm with Dynamic Gateway Selection and Delay Differentiation (CLC_DGS_DD) were proposed. They integrate an optimal traffic splitting scheme into a data transmission control framework with rate control, routing and scheduling. The CLC_DGS and CLC_DGS_DD algorithms have shown the ability of distributing traffic of a flow into multiple gateways in order to guarantee maximum network utility which is an increasing concave function of average rate. Besides achieving maximum network utility, CLC_DGS_DD is able to provide a flexible framework for adjusting delays among different flows, thus achieving low delays for preferential flows. All aforementioned studies were designed for data traffic in order to optimize network-oriented metrics. Therefore, there is a need of gateway selection for video streaming that can optimize user experience; hence the study of this paper.

2.2. Distributed Cooperative Algorithms and AD^3

To distribute a problem, a well-known method is Alternating Direction Method of Multipliers (ADMM) [32]. The major advantage of ADMM is its adaptability in various large-scale distributed problems. However, the slow convergence of ADMM prevents its application to dynamic wireless networks.

Recently, a new ADMM-based algorithm, the so-called AD^3 (Alternating Directions Dual Decomposition) has been proposed in the realm of the machine learning literature [20]. Besides obtaining faster convergence speed than ADMM, AD^3 has further interesting features as compared to other message-passing algorithms in the machine learning literature: it reaches consensus faster than other algorithms such as Tree-Reweighted Belief Propagation (TRBP) [33], or Projected Subgradient Dual Decomposition (PSDD) [34]; it neither has the convergence problems of Max-Product Linear Programming (MPLP) [35] nor the instability problems of Norm-Product Belief Propagation [36]; and its anytime design allows to stop the optimization process whenever a pre-specified accuracy is reached. Furthermore, as reported in [20], AD^3 has been empirically shown to outperform state-of-the-art message-passing algorithms on large-scale problems. Besides these features, AD^3 also provides a library of computationally-efficient factors that allow to handle hard constraints within an optimization problem (e.g. as shown in [37]). This opens the possibility of employing AD^3 to approximate constrained optimization problems in general. Notice that this goes beyond approximating the *Maximum a Posteriori* (MAP), which is the core problem tackled by the above-mentioned message-passing algorithms. Notice also that message-passing algorithms, such as AD^3, have been shown to outperform modern Linear Programming (LP) solvers such as CPLEX (e.g. [38]) in approximating large-scale MAP problems in a wide variety of application domains (e.g. computer vision, natural language processing). To the best of our knowledge, this is the first study on the application of AD^3 to dynamic wireless networks.

In practice, AD^3 is an iterative three-step algorithm designed to approximate an objective function encoded as a special graph-based structure, a so-called *factor graph*. A factor graph contains two types of nodes: factors (to represent the objective function and constraints) and variables (representing decision variables). Each factor is linked to its variables by means of edges. A key aspect of AD^3 is that it separates the optimization problem into independent sub-problems that progress to reach consensus on the values to assign to primal and dual variables. Thus, during the first step, the optimization problem is split into separate sub-problems, each one being distributed to a factor. Thereafter, each factor locally solves its local sub-problem. During the second step, each variable gathers the sub-problems' solutions of the factors it is linked to. Finally, during the third step, the Lagrange multipliers for each sub-problem are updated.

Employing AD^3 to approximate an optimization problem poses several challenges. A first challenge is to represent an objective function by means of a factor graph that solely contains computationally-efficient factors. This must be done to guarantee the fast computation of the messages exchanged by AD^3.

Furthermore, given that AD^3 solves an LP relaxation of an optimisation problem encoded as a factor graph, a further challenge is to design a distributed *decoding algorithm* that builds a feasible solution from the solution to the relaxed problem obtained by AD^3. Finally, a final, and fundamental challenge in this paper, is to run AD^3 in a distributed environment as we require. Notice that although AD^3 is amenable to parallelization [20], it has been mostly employed in a centralized manner in existing works.

3. The protocol

The outline of our proposed scheme is discussed in this section. First, the modifications of control packets of a conventional routing protocol in order to monitor network status are discussed. It is followed by a description of the steps of our proposed scheme.

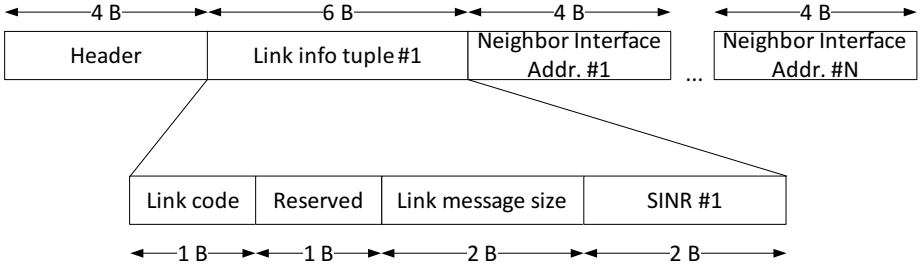
3.1. OLSR-based Protocol

In this paper, AD^3 is executed through collaboration between GWs. Each GW has to be aware of availability and quality of links (via SINR) in the networks to solve the optimization problem using AD^3. Introducing new packets to handle this task will increase the complexity and overhead. Consequently, we exploit existing control packets of the well-known *Optimized Link State Routing Protocol* (OLSR), i.e. `Hello` and `TC`. With this aim, in what follows we describe how to slightly modify the `Hello` and `TC` control packets.

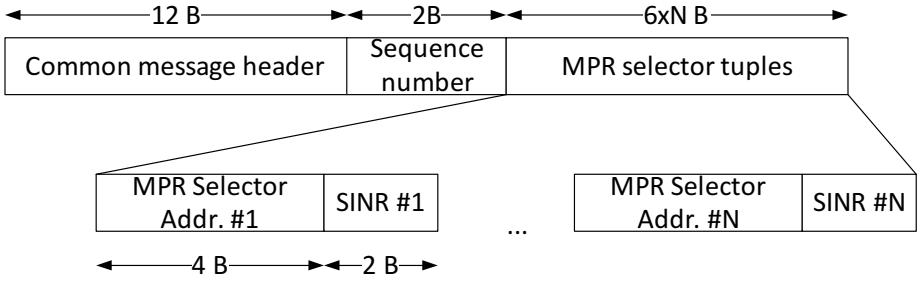
The proposed scheme adopts the control plane of OLSR. As mentioned above, the GWs have to monitor the status of the links in the network and this could be done by slightly modified control packets of OLSR. In OLSR, every node broadcasts `Hello` packets periodically. `Hello` packets contain information on state of links and neighbor interface addresses. Due to the exchange of `Hello` packets, each node can know about all its one-hop and two-hop neighbors and determine the multipoint relays (MPRs) from the set of one-hop neighbors. The MPRs are chosen so as to be able to reach every two-hop neighbor. After that, the lists of MPRs are broadcasted to one-hop neighbors using `Hello` packets. Then, each node creates the set of MPR Selectors and broadcasts it to other nodes in the networks using Topology control (`TC`) packets. Each node utilizes the information in `TC` packets to calculate the route to different destinations in the networks. Consequently, the gateways, which are also network nodes, are able to collect the network topology information through periodical control messages. Similar to [39, 40], the `Hello` and `TC` packets should be modified so as to be able to convey SINR information. The modification of control packets is described in Fig. 1.

3.2. Phases

We propose a three-phase scheme as described in Fig. 2.



(a) Modified `Hello` packet



(b) Modified `TC` packet

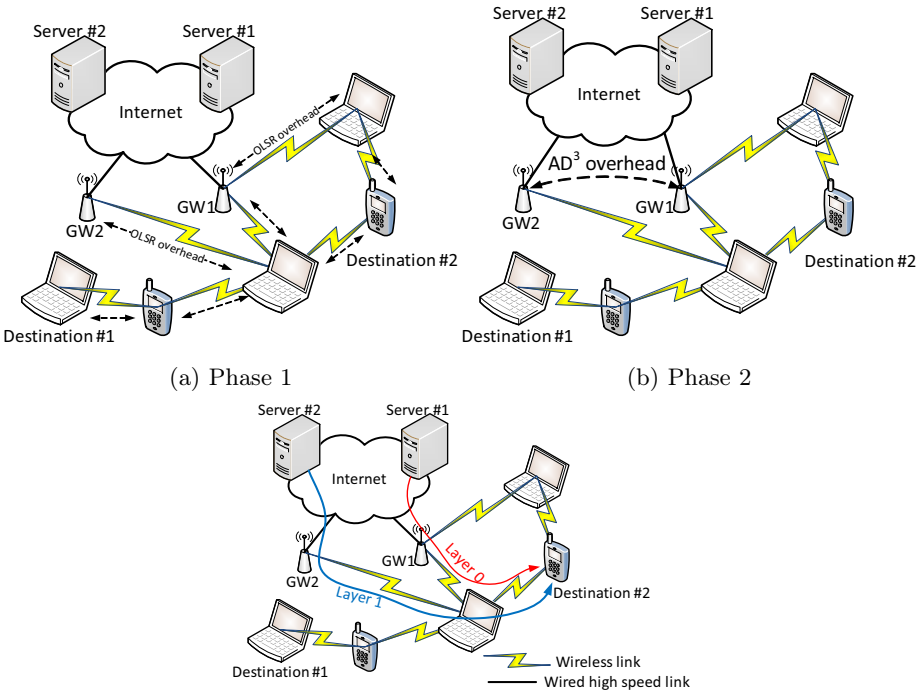
Figure 1: Modified control packets

Phase 1: Collect status of links through control packets of OLSR. In this phase, each gateway collects status of different links by receiving OLSR's control packets. Each gateway then forms a list of nodes it is able to reach. Then, the GWs exchange their list of nodes and links among themselves, in order to create factors.

Phase 2: Run AD^3 between gateways. The GWs cooperate to run AD^3. The number of iterations is predefined. Local solutions of GWs can be exchanged through the high speed connections. The variable and factor assignment problem will be discussed in Section 5.3 with an objective to minimize the number of messages exchanged.

Phase 3: Decoding and Streaming. The output of AD^3 is decoded by the Gateway-Layer Mapping Algorithm (GLAM) proposed in the following section. Then, each GW is aware of the layers of streams it is responsible for streaming.

Phase 1 runs in background as a part of standard OLSR. Meanwhile, Phase 2 will be triggered every τ seconds. As we should rerun AD^3 in case a change in topology is detected, the value of τ is chosen equal to the `TC` interval. The resulting routing algorithm is called AD^3-GLAM. Since heavy overhead in wireless networks is not desirable, it is unfeasible that all the nodes in the network cooperatively run AD^3. Additionally, the authors in [41] confirmed that a partially



(a) Phase 1

(b) Phase 2

(c) Phase 3

Figure 2: Phases of our proposed scheme

distributed mechanism offers a better solution quality and less overhead than a fully decentralized mechanism. Therefore, Phase 2 is done through cooperation between gateways instead of between all nodes in the networks.

3.3. Signal to Interference and Noise Ratio Prediction

In phase 1, the GWs collect the status of links through control packets of OLSR so as to form the optimization problem. The capacity of a link is inferred from its Signal to Interference and Noise Ratio (SINR). This section discusses the projection of the status of a link based on historical data.

In this paper, we adopt the physical interference model to depict the interference suffered by links in the network. In this model, by definition, communication between nodes is successful when the SINR (signal to interference and noise ratio) at the receiver is above a threshold θ. Note that this threshold depends on the desired characteristics of transmissions. Let us denote the received signal strength of a packet from node i to node j as $P_{i\to j}$. So, the SINR of link l from i to j is defined as follows

$$SINR_l = \frac{P_{i\to j}}{P_{\mathcal{N}} + \sum\limits_{k\neq i} P_{k\to j}}, \tag{1}$$

MCS	0	1	2	3	4	5	6
SINR(dB)	5.0	7.8	12.3	14.0	19.0	21.7	24.0

Table 1: MCS and SINR mapping [42]

where $P_{\mathcal{N}}$ is the background noise and $\sum_{k \neq i} P_{k \to j}$ is the total interference impacting the link from i to j. In 802.11n, the PER (packet error rate) depends on the SINR as well as the MCS (modulation and coding scheme) used. In other words, it means that the data rate is a function of SINR for a given PER. The capacity of a link can be determined as $c_l = f\left(SINR_l\right)$. For example, the mapping between SINR and MCS in order to obtain PER $< 10\%$ can be found in Table 1.

Time is divided into τ-second cycles. During a cycle, each node measures the SINR of its local links and sends this information to a gateway where the sub-optimal solutions are determined using a cooperative distributed mechanism. Let us denote $\sigma_l(t)$ as the average SINR of link l in cycle t. In this paper, we adopt the window mean exponentially weighted moving average (WMEWMA) estimator [43] for SINR estimation. The main reason we choose WMEWMA is its efficiency and simplicity. The predicted SINR of link l in cycle $(t+1)$, $\hat{\sigma}_l\left(t+1\right)$, can be determined as

$$\hat{\sigma}_l\left(t+1\right) = \left(1-\alpha\right)\sigma_l\left(t\right) + \frac{\alpha}{T} \sum_{k=\max(1,t-T)}^{t-1} \hat{\sigma}_l\left(k\right), \tag{2}$$

where T is the window length and $\alpha \in [0,1]$ is the adjustable weighting coefficient. The value of α is selected so as to minimize the prediction error. Note that when $T = 1$ this estimation is similar to the EWMA estimator proposed in [44].

Next, Section 4 formulates the optimization problem that will be solved during Phase 2. Section 5 details how to encode the optimization problem and optimize the number of exchanged messages while Section 6 details the decoding algorithm of Phase 3. We do not detail AD^3 since it consists in running AD^3 without modification, which is out of our contributions. The background of AD^3 is already discussed in Section 2.2

4. Problem Formulation

We consider a mesh wireless networks with multiple GWs. The GWs are fixed nodes with wired high speed connections and act as wireless access points. Meanwhile, other nodes are assumed to be equipped with 802.11n wireless cards. These other nodes are considered to be mobile with low to moderate speed ($0m/s$ to $3m/s$). A video model of scalable video coding is assumed in which video is hierarchically encoded into different layers. The combination of multiple layers can enhance the quality of received videos gradually as a function of the number of layers received. A layer of requested video can be streamed to a destination

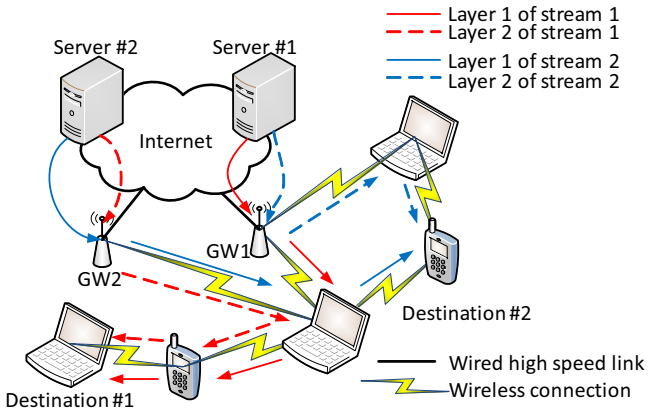


Figure 3: Wireless mesh networks with multicommodity flow

from a single gateway. However, a multiple-layer video can be downloaded from multiple GWs.

4.1. The Network and QoE Models

The network model and QoE models are going to be discussed in this section. We adopt the *multicommodity flow model* to formulate the routing problem in WMNs [45]. In this model, each flow is identified by its destination. Hence, the flows with the same destination are considered as a commodity, regardless of their gateways. Fig. 3 depicts an example of video streaming over WMNs with multiple gateways. Destinations 1 and 2 request the videos from the two servers. Layer 1 of stream 1 is downloaded from server 1 and layer 2 from server 2. A layer of a video traverses through a given end-to-end single path. However, the complete video arrives to the destination through multiple end-to-end paths.

We model the network using a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. The set of nodes $\mathcal{N}$ consists of N nodes, labeled $n = 1, ..., N$. They can send, receive, and relay data from sources to sinks. The set of links $\mathcal{A}$ comprises L directed links, labeled $l = 1, ..., L$. Let $\mathbb{O}(n)$ and $\mathbb{I}(n)$ be the sets of outgoing and incoming links of node n.

We denote destinations and sources as $d = 1, ..., D$ and $s = 1, ..., S$, where $D \leq N$ and $S \leq N$. The set of sources and destinations are denoted by $\mathbb{S}$ and $\mathbb{D}$, respectively. The node index of destination d is $\phi_d \in \mathcal{N}$. An integer source-sink vector $\mathbf{s}^{(d)}$ is defined for each destination d. When $n \neq \phi_d$, the entry $s_{n,k}^{(d)}$ is 1 if node n is the originator of layer k of stream d. When $n = \phi_d$, then $s_{n,k}^{(d)} = -1$ if layer k is received. Otherwise, $s_{n,k}^{(d)} = 0$. For the sake of clarification, definition of notations can be found in Table 2.

4.1.1. QoE model

We reuse the QoE model of SVC discussed in [15]. We denote γ_m as the maximum bit rate of m-layer videos. The maximum number of available layers

Notation	Definition
$x_{l,k}^{(d)}$	binary variable indicates layer k of stream d is conveyed by link l
$s_{n,k}^{(d)}$	binary variable indicates layer k of stream d originates at node n
γ_m	the required bandwidth of layer m
q_m	MOS of m-layer video
$\mathbb{I}(n)$	the set of incoming links at node n
$\mathbb{O}(n)$	the set of outgoing links at node n
$\mathbb{L}(n)$	the set of local links of node n, $\mathbb{L}(n) = \mathbb{I}(n) \cup \mathbb{O}(n)$
$\mathbb{S}$	the set of gateways
$\mathbb{D}$	the set of destinations
$\Psi^{(d)}$	MOS of stream d
$y_{i,j}$	binary variable indicates variable i belongs to node j
$z_{\alpha,j}$	binary variable indicates factor α belongs to node j
$b_{i,\alpha}$	binary variable indicates the connection between variable i and factor α
$\chi_{l,k}^{(d)}$	the cost of link l of layer k of stream d

Table 2: Definition of notations

is M. Once the network is able to admit stream d with bandwidth greater or equal to γ_m, m layers can be transmitted. Since MOS is a non-decreasing function of the number of layers, $\gamma_0 < \gamma_1 < ... < \gamma_M$ corresponds to QoE levels $q_0 < q_1 < ... < q_M$. Note that $(\gamma_0 = 0, q_0 = 1)$ means that no layer can be transmitted. Note that the number of layers and how to select the set of layers are out of the scope of this paper.

The quality of stream d can be defined as follows.

$$\Psi^{(d)} = q_0 + \sum_{n \in \mathbb{S}} \sum_{k=1}^{M} s_{n,k}^{(d)} \Delta_k \tag{3}$$

where $\Delta_k = q_{k+1} - q_k > 0$.

A layer of a video should be injected into the network from a single gateway. This can be formulated using the following constraint

$$\sum_{n \in \mathbb{S}} s_{n,k}^{(d)} \leq 1, \forall k \tag{4}$$

Moreover, the layer $k+1$ is transmitted only if the layer k was also transmitted, which is

$$\sum_{n \in \mathbb{S}} s_{n,k}^{(d)} \geq \sum_{n \in \mathbb{S}} s_{n,k+1}^{(d)}, \forall k \tag{5}$$

4.1.2. Network model

We denote binary variable $x_{l,k}^{(d)}$ as the indicator that layer k of stream d is conveyed by link l. If link l is utilized to convey layer k of stream d, $x_{l,k}^{(d)} = 1$. Otherwise, $x_{l,k}^{(d)} = 0$. We have

$$\sum_{l \in \mathbb{O}(n)} x_{l,k}^{(d)} - \sum_{l \in \mathbb{I}(n)} x_{l,k}^{(d)} = s_{n,k}^{(d)} \tag{6}$$

To avoid negative value of $s_{n,k}^{(d)}$ when $n = \phi_d$, the above equation is modified as follows

$$\sum_{l \in \mathbb{I}(\phi_d)} x_{l,k}^{(d)} - \sum_{l \in \mathbb{O}(\phi_d)} x_{l,k}^{(d)} = s_{\phi_d,k}^{(d)} \tag{7}$$

In $\mathbb{I}(n)$ and $\mathbb{O}(n)$, there is at most one link conveying a layer of a video. That means

$$\begin{aligned} \sum_{l \in \mathbb{I}(n)} x_{l,k}^{(d)} &\leq 1 \\ \sum_{l \in \mathbb{O}(n)} x_{l,k}^{(d)} &\leq 1 \end{aligned} \tag{8}$$

The total amount of time a node spends to relay a layer k to destination d is formulated by time-constraint model proposed in [46]. This model has been widely used in works such as [12, 47].

$$\sum_{l \in \mathbb{L}(n)} \sum_{d=1}^{D} \sum_{k=0}^{M} \frac{x_{l,k}^{(d)} \left(\gamma_k - \gamma_{k-1}\right)}{c_l} \leq \rho, \forall n \tag{9}$$

where ρ should be $< \frac{2}{3}$ for MAC protocol feasibility [48]. The real payload, however, may occupy only 50% of available transmission time in interference environment [49] because of the collisions and re-transmissions. In fact, ρ can be selected dynamically according to interference. However, the selection of ρ value is not in the scope of this paper. We select $\rho = 0.5 \times \frac{2}{3} = \frac{1}{3}$ in this paper. The capacity of link c_l can be inferred from SINR.

4.1.3. The Optimization Problem

The objective is thus to solve the following problem.

Problem 1 (QoE-based Routing Problem for SVC)**.** *Our problem amounts to maximizing the total MOS of all users in the network:*

$$\begin{aligned} \max \quad & \sum_{d \in \mathcal{D}} \Psi^{(d)} \\ s.t. \quad & (4), (5), (6), (7), (8), (9) \\ & x_{l,k}^{(d)}, s_{n,k}^{(d)} \in \{0, 1\} \end{aligned} \tag{10}$$

To analyze the complexity of problem 1, we introduce the generalized maximum coverage problem.

Definition 1 (Generalized Maximum Coverage problem (GMC) [50])**.** *Given a budget L, a set $\mathcal{E}$ of elements, a set of bins $\mathcal{B}$, a positive profit $P(b,e)$ and a non-negative weight $W(b,e)$ for each tuple (b,e), and overhead of using bin b as $W(b)$, find a triple $S=(\beta,\eta,f)$, where $\beta \subseteq \mathcal{B}, \eta \subseteq \mathcal{E}$, and f is an assignment function from η to β guaranteeing that each element e is assigned to a unique bin b. The weight of selection is $W(S)=\sum_{b\in\beta} W(b)+\sum_{e\in\eta} W(f(e),e)$. This weight is limited by the budget L, such that $W(S)\leq L$. The profit of selection is $P(S)=\sum_{e\in\eta} P(f(e),e)$.*

Theorem 1. *For any $\epsilon > 0$, Problem 1 has no $(1-1/e+\epsilon)$ approximation algorithm unless $P=NP$.*

Proof. We show that a special case of Problem 1 is equivalent to generalized maximum coverage problem (GMC).

Let us consider a special case of our Problem 1, where there are N_D+2 nodes containing a source, a relaying node, and N_D destinations as shown in Fig. 4. The video is encoded into M layers. Let us denote m as the total number of layers received at the destination and $\mathcal{M}$ is the set of available values of m. Now, this special case of Problem 1 is equivalent to GMC. Indeed, $\mathcal{M}$ and $\mathcal{D}$ are equivalent to $\mathcal{B}$ and $\mathcal{E}$ in GMC. The number of total received layers at a destination is unique, so function f of GMC is automatically satisfied. At the relaying node, the utilization, in terms of node occupancy in time, to forward layer k of stream r is

$$\tau(k,d)=\gamma_k\left(\frac{1}{c_{s,r}}+\frac{1}{c_{r,d}}\right) \tag{11}$$

Note that the node occupancy constraint at the relaying node consists of all the other node occupancy constraints. Further, the weight of tuple (b,e) in GMC corresponds to $\tau(k,d)$. Overhead of using bin b is 0. Consequently, the budget L in GMC corresponds to ρ in Problem 1. The profit $P(b,e)$ in GMC will be $P(k,d)=q_k$ which is the MOS of stream d. Thus, this special case of Problem 1 can be directly mapped to GMC. In [50], the authors showed that the upper-bound approximation ratio of GMC is $\frac{e}{e-1}$ since it holds MC as a special case. Moreover, GMC is a NP-hard problem. The special case of Problem 1 has one-to-one relationship with the GMC problem, so Problem 1 is at least as hard as GMC. □

5. Encoding and solving the optimization problem

In the previous section, we discussed the need for AD^3 and the modifications of the OLSR routing protocol. AD^3 requires that the optimization problem posed by Problem 1 is encoded as a factor graph. The encoding and solving processes are executed during the phase 2 described in Section 3.2, and hence right after phase 1 updates the global status of the network through OLSR TC.

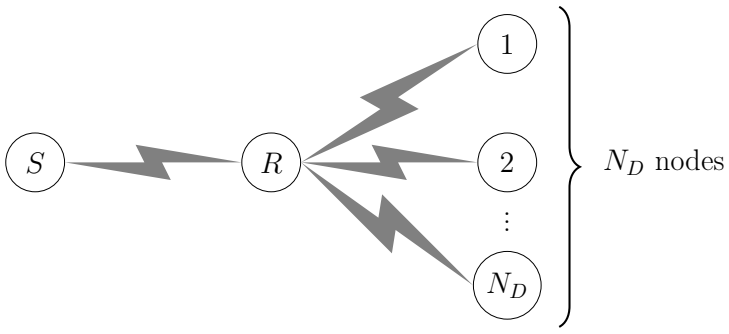


Figure 4: Special case of the problem

The information exchange required to run AD^3 can be done through reliable Transmission Control Protocol (TCP) over a high speed wired network. Each gateway knows the variables and factors that it controls as a result of solving the factor and variable assignment problem (problem 2 in Section 5.3), which distributes the factor graph among gateways. Once AD^3 converges to a solution, we need a decoding algorithm in order to convert it to a feasible solution. In this section, we discuss how to encode the optimization problem as a factor graph.

5.1. AD^3-based optimization

Problem 1 is a NP-hard problem. Although it can be solved by state-of-the-art integer linear programming solvers, it is unfeasible to use them to solve large-scale problems because of the limitations of computation resources and calculation time. As we mentioned in Section 2, AD^3 is able to provide faster convergence than ADMM and has a library of computationally-efficient factors to tackle hard constraints in an optimization problem. The solution obtained by AD^3 may not be a feasible solution because AD^3 solves a relaxation of the optimization problem. However, it can be exploited to derive a feasible solution close to the optimal one by a decoding algorithm like the one we present in Section 6.

Running AD^3, nevertheless, requires synchronization between GWs. The benefits of cooperation between mediators (GWs in this paper) have been discussed in [41]. AD^3 will shift to next iteration when all nodes finish solving their local problems and exchanging solutions. Our algorithm runs over an existing routing protocol with some slight modifications which will be presented in the following section.

5.2. Encoding the Optimization Problem

For the sake of readability, we provide an example in order to demonstrate the whole encoding procedure of the problem.

Example 1. *Let us consider a network with* 4 *nodes as shown in Fig. 5, where node* 0 *and node* 1 *are gateways, node* 2 *is a relaying node, and node* 3 *is a*

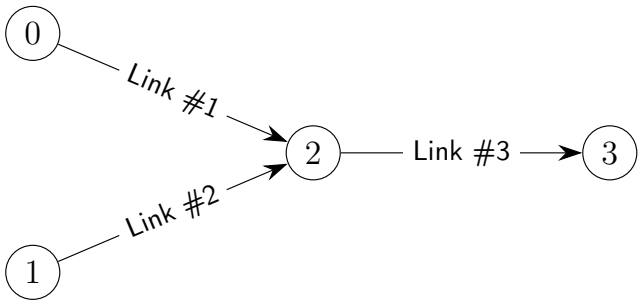


Figure 5: Sample network from Example 1

destination node. For simplicity, we consider a video with two layers 1 *and* 2*, corresponding to the MOS values* 2.451 *and* 2.748*. Note that with 2 layers we will have delta MOS values* (Δ_k) *of* 1.451 *and* 0.297*. Delta MOS value is the difference of MOS values of a given layer and its previous layer. The difference between maximum bit rate between two layers is* 0.23 *Mbps. The air-time constraint at node* 2 *already comprises the air-time constraints of other nodes, in case it comes under their transmission ranges. Thus, we consider only the air-time constraint at node* 2*. Following the definition of problem 1, the QoE-based routing problem for SVC can be encoded as the following integer linear program (ILP):*

$$
\begin{array}{ll}
\max & 1.0 + 1.451\,(s_{0,1} + s_{1,1}) + 0.297\,(s_{0,2} + s_{1,2}) \\
s.t. & s_{0,1} + s_{1,1} \leq 1 (cs_1),\ s_{0,2} + s_{1,2} \leq 1 (cs_2) \\
 & s_{0,1} + s_{1,1} \geq s_{0,2} + s_{1,2} (cs_3) \\
 & x_{1,1} = s_{0,1} (cs_4),\ x_{1,2} = s_{0,2} (cs_5) \\
 & x_{2,1} = s_{1,1} (cs_6),\ x_{2,2} = s_{1,2} (cs_7) \\
 & x_{3,1} = s_{3,1} (cs_8),\ x_{3,2} = s_{3,2} (cs_9) \\
 & x_{1,1} + x_{2,1} - x_{3,1} = 0 (cs_{10}) \\
 & x_{1,2} + x_{2,2} - x_{3,2} = 0 (cs_{11}) \\
 & \frac{x_{1,1}+0.23x_{1,2}}{c_1} + \frac{x_{2,1}+0.23x_{2,2}}{c_2} + \frac{x_{3,1}+0.23x_{3,2}}{c_3} \leq \rho (cs_{12}) \\
 & x_{l,k}, s_{n,k} \in \{0, 1\}
\end{array}
\tag{12}
$$

Each variable contributes to the global MOS. From the objective function, the variables $s_{0,1}, s_{1,1}, s_{0,2}, s_{1,2}$ *contribute positively to the global MOS, while others are assigned zero value in terms of their contribution. There are 12 constraints in the above ILP, which are indexed from* cs_1 *to* cs_{12}*. Constraints* cs_1 *and* cs_2 *are derived from Eq.* (4)*. Eq.* (5) *is captured by* cs_3*. The constraints from* cs_4 *to* cs_{11} *are based on the routing constraints in Eq.* (6) *and* (7)*. The constraints in Eq.* (8) *are automatically satisfied, and hence are discarded. Constraint* cs_{12} *is the air-time constraint at the relaying node. Fig. 6 shows the factor graph encoding the example. Eclipse-shaped objects represent variables and the inner fractional value is the contribution of the variable to the objective function. Meanwhile, rectangular shaped objects are the factors corresponding to constraints. A link between a variable and a factor exists when that variable is*

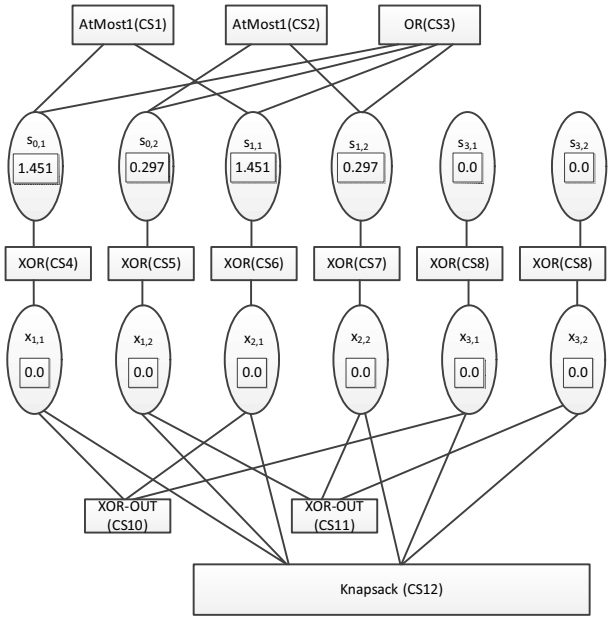


Figure 6: Factor graph of Example 1

related to the constraint represented by the factor.

In order to adopt AD^3, we have to encode our optimization problem into a factor graph. Moreover, recall from Section 2 that our aim is to solely employ computationally-efficient factors, so that the computation of AD^3 as well as the use of messages is efficient. Next, we introduce the factors (functions) from [20] that will allow the encoding of the constraints of Problem 1.

Definition 2 (**OR** factor)**.** *It represents a disjunction of K binary variables ($K \geq 1$) defined through the following potential function*

$$\theta_{\mathbf{OR}}(x_1, ..., x_K) \coloneqq \begin{cases} 0 & \text{if } x_1 \vee x_2 \vee ... \vee x_k = 1 \\ -\infty & \text{otherwise} \end{cases} \tag{13}$$

Definition 3 (**AtMost1** factor)**.** *It constrains at most one of the variables $x_1, \ldots, x_K$ to be active. Its potential function is defined as:*

$$\theta_{\mathbf{AtMost1}}(x_1, ..., x_K) \coloneqq \begin{cases} 0 & \text{if } \exists!\ k \text{ s.t. } x_k = 1 \\ & \vee x_1 = ... = x_K = 0 \\ -\infty & \text{otherwise} \end{cases} \tag{14}$$

Definition 4 (**XOR** factor)**.** *It constrains that exactly one of the variables $x_1, \ldots, x_K$ takes value 1 through the potential function:*

$$\theta_{\mathbf{XOR}}(x_1, ..., x_K) \coloneqq \begin{cases} 0 & \text{if } \exists!\ k \text{ s.t. } x_k = 1 \\ -\infty & \text{otherwise} \end{cases} \tag{15}$$

Definition 5 (**XOR-out** factor)**.** *It constrains at most one of the variables $x_1, \ldots, x_K$ to be active; if one is active, it constrains $x_{K+1} = 1$; if all are inactive, then it constrains $x_{K+1} = 0$. Its potential function is defined as:*

$$\theta_{\mathbf{XOR-out}}(x_1, \ldots, x_K, x_{K+1}) \coloneqq \begin{cases} 0 & \text{if } x_{K+1} = 1 \wedge \nexists k \in \{1, \ldots, K\} : x_k = 1 \\ 0 & \text{if } x_{K+1} = 0 \wedge \forall k \in \{1, \ldots, K\} : x_k = 0 \\ -\infty & \text{otherwise} \end{cases} \tag{16}$$

Definition 6 (Knapsack (**KS**) factor)**.** *Its potential function can be defined as:*

$$\theta_{\mathbf{KS}}(x_1, ..., x_K) \coloneqq \begin{cases} 0 & \text{if } \sum_k x_k \leq C \\ -\infty & \text{otherwise} \end{cases} \tag{17}$$

where C is a given constant.

Now we can encode the above optimization problem into a factor graph as illustrated in Figure 6. Variables, represented as round circles, are linked to the factors representing the hard constraints in the problem, which in turn are represented as rectangles. Each variable contains the value obtained when the variable is active. For instance, $s_{0,1}$ contains 1.451. In this way we encode our objective function.

In general, the optimization Problem 1 can be encoded as follows: Eq. (4) and (8) can be encoded by using the **AtMost1** factor; Eq. (5) can be rewritten as $\sum_{n\in\mathbb{S}} s^{(d)}_{n,k} - \sum_{n\in\mathbb{S}} s^{(d)}_{n,k+1} \geq 0$, then it can be encoded by an **OR** factor; and Eq. (9) can be encoded by a **KS** factor. In order to encode Eq. (6) and (7), we create auxiliary variables $x^{(d)}_{in,k} = \sum_{l\in\mathbb{I}(n)} x^{(d)}_{l,k}$ and $x^{(d)}_{out,n,k} = \sum_{l\in\mathbb{I}(n)} x^{(d)}_{l,k}$. By Eq. (8), $x^{(d)}_{in,n,k}$ and $x^{(d)}_{out,k}$ is in $\{0, 1\}$. Then, Eq. (6) and (7) can be rewritten as $x^{(d)}_{out,n,k} = x^{(d)}_{in,n,k} + s^{(d)}_{n,k}$, and hence can be described by an **XOR-out** factor. When a node is not a gateway $\left(s^{(d)}_{n,k} = 0\right)$ or does not have either incoming $\left(x^{(d)}_{in,n,k} = 0\right)$ or outgoing links $\left(x^{(d)}_{out,n,k} = 0\right)$, Eq. (6) and (7) will turn into $x^{(d)}_{in,n,k} - x^{(d)}_{out,n,k} = 0$, $x^{(d)}_{in,n,k} - s^{(d)}_{n,k} = 0$, or $x^{(d)}_{out,n,k} - s^{(d)}_{n,k} = 0$ respectively. All of them can be encoded by means of **XOR** factors.

Following [20], the complexity of the **OR**, **XOR**, **AtMost1**, and **XOR-out** factors is $O(K \cdot logK)$, where K stands for the number of variables connected to the **XOR** factor. Moreover, according to [51], the complexity of the **KS** factor is linear with the size of the factor. Therefore, we have managed to provide an encoding of our optimization problem that only employs computationally-efficient factors.

5.3. Factor and Variable Assignment Problem

In what follows, we detail how to distribute a factor graph, which encodes our optimization problem, between the gateways in a network. In this way, we will be able to run AD^3 in a distributed manner between GWs. Note that GWs

are the sources of the streams in this paper. As OLSR is a link-state routing algorithm, each GW maintains a database of link-state obtained by receiving control packets. Let us denote $\mathcal{L}_s$ as the set of links which are in the database at the gateway s. The number of exchanged messages can be minimized by solving a joint variable and factor assignment problem. Let us denote binary variables $y_{i,j} \in \{0,1\}$ and $z_{\alpha,j} \in \{0,1\}$ as follows

$$y_{i,j} = \begin{cases} 1 & \text{if variable } i \text{ belongs to node } j \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

$$z_{\alpha,j} = \begin{cases} 1 & \text{if factor } \alpha \text{ belongs to node } j \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Note that $y_{i,j} = 0$ if $i \notin \mathcal{L}_j$. Let us denote the binary variable $b_{i,\alpha}$ so that $b_{i,\alpha} = 1$ if and only if there is a connection between variable i and factor α. We assume that a message is created when a factor requests the value of a variable from another node. The total number of messages is

$$\sum_{i,\alpha} b_{i,\alpha} \left(\sum_{s \in \mathbb{S}} (1 - y_{i,s} z_{\alpha,s}) \right) \tag{20}$$

where $\mathbb{S}$ is the set of sources (GWs). We define an auxiliary binary variable $\tilde{z}^s_{i,\alpha} = y_{i,s} \cdot z_{\alpha,s}$, then

$$\tilde{z}^s_{i,\alpha} \leq y_{i,s} \tag{21}$$

$$\tilde{z}^s_{i,\alpha} \leq z_{\alpha,s} \tag{22}$$

$$\tilde{z}^s_{i,\alpha} \geq y_{i,s} + z_{\alpha,s} - 1 \tag{23}$$

Moreover, each variable and factor should be assigned to one of the GWs by enforcing the following constraints:

$$\sum_{s \in \mathbb{S}} y_{i,s} \leq 1 \tag{24}$$

$$\sum_{s \in \mathbb{S}} z_{\alpha,s} \leq 1 \tag{25}$$

Consequently, we have an optimization problem to efficiently distribute and assign a factor graph by minimizing the number of messages and considering the above constraints:

Problem 2 (Factor and Variable Assignment Problem)**.**

$$\begin{array}{ll} \min & \sum_{i,\alpha} b_{i,\alpha} \left(\sum_{s \in \mathbb{S}} \left(1 - \tilde{z}^s_{i,\alpha}\right) \right) \\ s.t. & (21), (22), (23), (24), (25) \\ & \tilde{z}^s_{i,\alpha}, y_{i,s}, z_{\alpha,s} \in \{0,1\} \end{array} \tag{26}$$

The above problem is an integer linear programming (ILP). Note that the complexity of this problem is much smaller than that of Problem 1, thus it can be solved by using off-the-shelf ILP solver such as CPLEX or Gurobi [52, 53]. An approximate algorithm for this problem may be necessary to cope with large-scale problems, however it is out of scope of this study.

6. Heuristic Decoding Algorithm

AD^3, as explained in Section 2, is used to solve a relaxed version of Problem 1. Practically, the algorithm may have to stop after M iterations because of the limitations in terms of calculation time. The solution, therefore, comprises fractional values which may not feasible for the optimization routing problem. Thus, below we describe a distributed decoding algorithm to obtain the final feasible solution.

6.1. Cost of Path

AD^3 may provide fractional values of some variables. Once we round them to obtain a feasible solution, we increase the link utilization and thus incur some additional cost called as cost of path. Besides, we may gain better MOS with this rounding and that should also be considered.

Let us denote $\tilde{x}_{l,k}^{(d)}$ and $\tilde{s}_{n,k}^{(d)}$ as the fractional solution of AD^3 to problem 1. First, we propose the cost of link based on the fractional solution of AD^3. The cost of link is defined for each tuple (l, k, d), where l is the link index, k is the layer, and d is the destination. Note that the link will be utilized to convey layer k of stream d when $x_{l,k}^{(d)} = 1$. The air-time cost of rounding a fractional $\tilde{x}_{l,k}^{(d)}$ to 1 is $\frac{\left(1-\tilde{x}_{l,k}^{(d)}\right)(\gamma_k-\gamma_{k-1})}{c_l}$. Also, the potential profit obtained by rounding link $\tilde{x}_{l,k}^{(d)}$ to 1 can be the gain in MOS $(q_k - q_{k-1})$ of stream d. Therefore, we define the cost of link as the ratio of the air-time cost to the potential profit as

$$\chi_{l,k}^{(d)} = \frac{\left(1 - \tilde{x}_{l,k}^{(d)}\right)(\gamma_k - \gamma_{k-1})}{c_l\,(q_k - q_{k-1})} \tag{27}$$

where c_l and γ_k are the capacity of link l and the bandwidth requirement of layer k respectively. The capacity of link l, c_l, is identified from the up-to-date `Hello` and `TC` packets. Therefore, in this way the decoding algorithm becomes aware of link quality changes. In particular, that helps the decoding algorithm to assign a high cost to bad links, thus avoiding their usage.

Indeed, the impact of link selection is local. Thus, the cost of path cannot be determined by summing up of all the costs along the path. Alternatively, it should be the maximum cost of involving such links

$$\chi_{\mathcal{P}_k^{(d)}} = \max_{l \in \mathcal{P}_k^{(d)}} \chi_{l,k}^{(d)}, \tag{28}$$

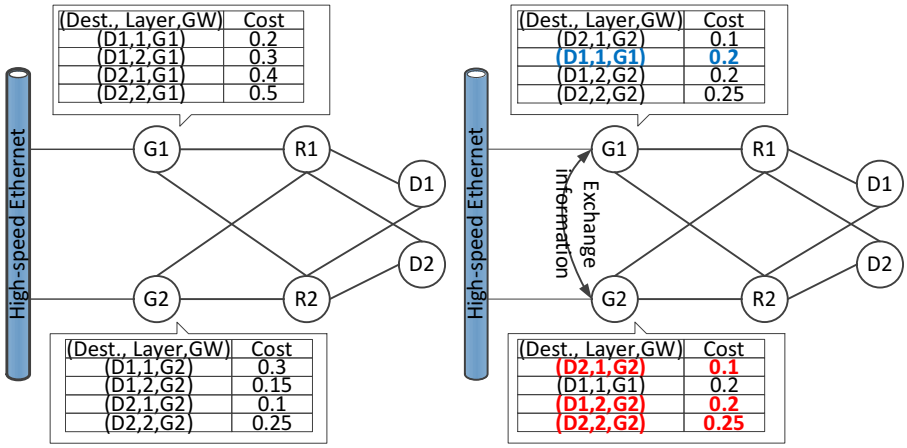
where $\mathcal{P}_k^{(d)}$ is the path to destination d of layer k. The optimal path of (k,d) is the path with minimal cost among available paths to (k,d). The algorithm to find the optimal path for layer k of stream d is described in Alg. 1. Line 6 to line 10 is the major part of the algorithm. Each GW finds the optimal path for layer k of stream d. Then, GWs exchange their solution to find out the gateway that provides the lowest cost path to stream layer k to destination d. If there exists a path from any GW to the destination, the lowest cost path will be output by the algorithm, as shown in lines 11 and 12. Otherwise, an empty set will be the output of the algorithm (line 14)

Algorithm 1: Finding optimal path for (d,k)

1 **Input:**Set of gateways $\mathbb{G}$, $\tilde{x}_{l,k}$ (output of AD^3)
2 **Output:** The optimal path for layer k of stream d - $\mathcal{P}_k^{*(d)}$ and the corresponding cost of path $\chi_{\mathcal{P}_k^{*(d)}}$ and gateway $\mathcal{G}_k^{*(d)}$
3 $\chi_{\mathcal{P}_k^{*(d)}} = \infty$;
4 $\mathcal{G}_k^{*(d)} = \emptyset$;
5 $\mathcal{P}_k^{*(d)} = \emptyset$;
6 **foreach** $g \in \mathbb{G}$ **do**
7 **if** *there is* $P_k^{(d)}$ *from* g ***and*** $\chi_{\mathcal{P}_k^{(d)}} < \chi_{\mathcal{P}_k^{*(d)}}$ **then**
8 $\chi_{\mathcal{P}_k^{*(d)}} = \chi_{\mathcal{P}_k^{(d)}}$;
9 $\mathcal{G}_k^{*(d)} = g$;
10 $\mathcal{P}_k^{*(d)} = P_k^{(d)}$;
11 **if** $\mathcal{P}_k^{*(d)} \neq \emptyset$ **then**
12 **return** $\left(\mathcal{P}_k^{*(d)}, \chi_{\mathcal{P}_k^{*(d)}}, \mathcal{G}_k^{*(d)}\right)$
13 **else**
14 **return** $\emptyset$

*6.2. Gateway-Layer Mapping Algorithm (*GLAM*)*

In this section, we are going to discuss the gateway-layer mapping algorithm (GLAM), or Alg.2, in details. This algorithm is able to decode AD^3 solution in a distributed manner. The objective of GLAM is to assign a video layer to a GW in order to maximize the number of transmitted layers. The algorithm can be divided into two main parts. The first part is from line 5 to line 11. The objective of the first part is to determine the availability of paths and the priority for each (d,k). In line 7, the optimal path and its corresponding gateway is determined by using Alg. 1, for each (d,k). At line 8, the algorithm checks the availability of the path. If the path exists, the cost of the path will be checked in line 9 and line 10. If the cost of the path of stream d and layer k is less than the cost of the path of layer $k-1$, then the new cost for the



(a) Each GW calculates optimal paths to destinations

(b) Exchange information between GWs and run Alg. 2

Figure 7: Example of decoding process

path that is equal to the cost of layer $k-1$ will be assigned to layer k. Then, the tuple $\left(d, k, \chi_{\mathcal{P}_k^{*(d)}}, \mathcal{G}_k^{*(d)}\right)$ is added into set $\mathbb{U}$. The priority of each (d,k) is determined based on the optimal cost of paths and its layer in line 12. The (d,k) pair with lower cost is assigned higher priority. When the cost is equal, the lower layer will have higher priority. The reassignment cost process in lines 9 and 10 guarantees that a layer will not be transmitted unless the lower layer was transmitted.

Note that the solution after finishing part 1 satisfies the routing and integer constraints of the original problem. In the second part, the air-time constraints are considered. From line 15 to line 25, the process of tackling paths violating air-time constraints is described. The links violating the air-time constraints will be assigned the cost equal to infinity, in order not to be chosen later. Then, Alg. 1 is applied to find a new optimal path for (d,k) in line 17. If the path exists, a process to check the cost and sort $\mathbb{U}$ will be triggered from line 18 to line 22. Otherwise, all layers which are greater or equal to k will be discarded (lines 24 and 25) because layer k is missing. The process for paths which do not violate the air-time constraints is described from line 26 to line 29. The layer k of stream d will be removed from $\mathbb{U}$ since this layer has already been considered. After that, the streaming process for layer k of stream d begins.

To demonstrate the decentralized operation of the decoding algorithm, we introduce an example described in Fig. 7. Two GWs, $G1$ and $G2$, connect to a high speed wired network and two destinations, $D1$ and $D2$, connect to GWs through relaying nodes $R1$ and $R2$. For simplicity, a two-layer video is considered. At the beginning, all GWs exchange their fractional solutions, which are the output of AD^3, and calculate the optimal paths to all destinations (Alg. 1). We assume the cost of optimal paths as shown in Fig. 7a. Then, GWs

Algorithm 2: Gateway-Layer Mapping Algorithm (GLaM)

Input: Set of streams $\mathbb{D}$, set of gateways $\mathbb{G}$
Output: Set of stream-layer $\mathbb{V}$ and corresponding gateways $\{\mathcal{G}_{k,d}\}$
$\mathbb{V} = \emptyset$;
$\mathbb{U} = \emptyset$;
foreach $d \in \mathbb{D}$ **do**
    foreach *layer* k **do**
        $\left(\mathcal{P}_k^{*(d)}, \chi_{\mathcal{P}_k^{*(d)}}, \mathcal{G}_k^{*(d)}\right) \leftarrow$ Find the optimal path for (d, k) (see Alg.1);
        if $\mathcal{P}_k^{*(d)} \neq \emptyset$ **then**
            if $\chi_{\mathcal{P}_k^{*(d)}} < \chi_{\mathcal{P}_{k-1}^{*(d)}}$ **then**
                $\chi_{\mathcal{P}_k^{*(d)}} = \chi_{\mathcal{P}_{k-1}^{*(d)}}$
            $\mathbb{U} \leftarrow \left(d, k, \chi_{\mathcal{P}_k^{*(d)}}, \mathcal{G}_k^{*(d)}\right)$;
Sort $\mathbb{U}$ in ascending order of $\chi_{P_k^{*(d)}}$ and k;
while $|\mathbb{U}| > 0$ **do**
    foreach *entry* $\left(d, k, \chi_{\mathcal{P}_k^{*(d)}}, \mathcal{G}_k^{*(d)}\right)$ *in* $\mathbb{U}$ **do**
        if $\mathcal{P}_k^{*(d)}$ *violates any air-time constraint* **then**
            $\chi_{l,k}^{(d)} \leftarrow \infty$ for links violating air-time constraints;
            Run Alg. 1 $\rightarrow \left(\mathcal{P}_k^{*(d)}, \chi_{\mathcal{P}_k^{*(d)}}, \mathcal{G}_k^{*(d)}\right)$;
            if $\mathcal{P}_k^{*(d)} \neq \emptyset$ **then**
                if $\chi_{\mathcal{P}_k^{*(d)}} < \chi_{\mathcal{P}_{k-1}^{*(d)}}$ **then**
                    $\chi_{\mathcal{P}_k^{*(d)}} = \chi_{\mathcal{P}_{k-1}^{*(d)}}$
                $\mathbb{U} \leftarrow \left(d, k, \chi_{\mathcal{P}_k^{*(d)}}, \mathcal{G}_k^{*(d)}\right)$;
                Sort $\mathbb{U}$ in ascending order of $\chi_{P_k^{*(d)}}$ and k;
            else
                foreach $m \geq k$ **do**
                    $\mathbb{U} \setminus \left(d, m, \chi_{P_m^{*(d)}}, \mathcal{G}_m^{*(d)}\right)$;
        else
            $\mathbb{U} \setminus \left(d, k, \chi_{\mathcal{P}_k^{*(d)}}, \mathcal{G}_k^{*(d)}\right)$;
            $\mathbb{V} \leftarrow \left(d, k, \mathcal{G}_k^{*(d)}\right)$;
            Start Streaming layer k for stream d from gateway $\mathcal{G}_k^{*(d)}$;

exchange these optimal path information through high speed network and run Alg. 2 as shown in Fig. 7b. First, they sort all entries in ascending order of the cost. Consequently, the first entry is $(D2, 1, G2)$ with the cost 0.1. Although the initial cost of the entry $(D1, 1, G1)$ is greater than the entry $(D1, 2, G2)$ (0.2 and 0.15), $(D1, 1, G1)$ still it has higher priority than $(D1, 2, G2)$ because of lines 9 and 10 in Alg. 2. By doing this, we guarantee compliance with constraints (5). The last entry is $(D2, 2, G2)$ with the cost 0.25 and the algorithm terminates.

7. Simulation Results

We evaluate the performance of the proposed mechanism using network simulator (NS) version 3.25 [54]. We consider a rural area where cellular networks may not be available. The terrain size is $500m \times 500m$. The number of mobile nodes is 15 to 35. There are four fixed GWs at $(125, 125)$, $(125, 375)$, $(375, 125)$, and $(375, 375)$. Mobile nodes move according to the random walk model with the arbitrary speed in range $[0, 3]\, m/s$. We adopt the log distance propagation loss model at $2.4GHz$ for physical layer simulation. In medium access control (MAC) layer, we adopt 802.11n standard with single spatial stream. The details can be found in Table 4.

Refer to [15], the number of layers is 7 for numerical analysis. Table 3 shows the mapping between MOS, maximum bit rate, QP, and FPS using PSQA tool [11].

m	**QP**	**FPS**	**Maximum bit rate (Mbps)**	**MOS** (q_m)
1	44	7.5	1.0	2.451
2	42	7.5	1.23	2.748
3	36	7.5	2.3	3.194
4	36	15	3.03	3.602
5	28	15	6.96	3.959
6	28	30	9.2	4.791
7	22	30	17.14	5

Table 3: QoE levels, Maximum bit rate, QP and FPS

7.1. Prediction Error

First, we conduct some simulations in order to determine the optimal α for the SINR estimator. We consider the most dense scenarios (35 nodes), which have heaviest interference among all scenarios. The window length is 1 , 5, and 10. Fig. 8 shows that the error is minimized with $\alpha = 0.6$ and $T = 5$.

7.2. Performance

We study the average MOS value with a varying number of streams and nodes. Each simulation configuration runs 30 times with different initial positions of mobile nodes.

Parameter	Value
Path loss exponent	3
Reference loss	40.046 dB
Channel Width	20 Mhz
Spatial streams	1
Transmission power	15 dBm
Short Guard Interval	No
RTS/CTS	No
Hello Interval	1s
TC Interval	5s

Table 4: Simulation parameters

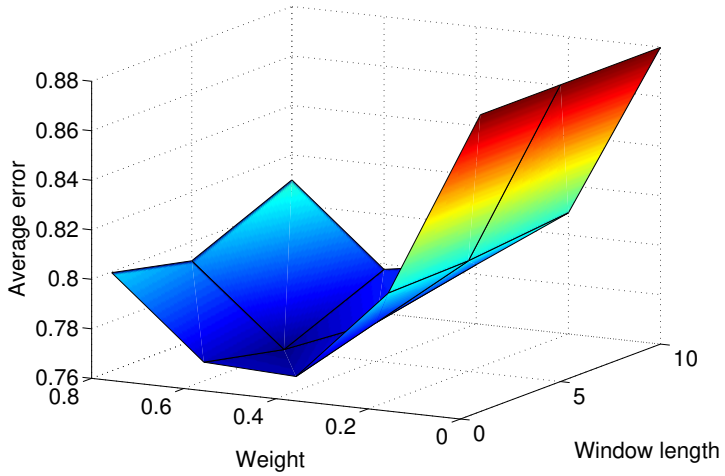


Figure 8: Average Error vs Weight (α) and Window Length (T)

7.2.1. MOS and calculation time

Fig. 9 shows that the average MOS generally degrades when the number of nodes decreases and the number of streams increases. The MOS value does not change significantly when varying the number of nodes. In fact, the increase in number of nodes can increase the number of connections in the network. However, the interference between links may prevent the capacity of networks from increasing. The degradation in MOS caused by interference also shows up when the number of streams increases. An increase in the number of streams leads to heavier traffic on links and more links start to get utilized. Hence, the interference issue becomes more severe.

The comparison of OLSR with AD^3-GLAM is shown in Fig. 10 and Fig. 11. Generally, the proposed mechanism offers better performance than OLSR. Furthermore, the increase in the number of iterations can enhance the performance of the proposed mechanism. In Fig. 10, the number of streams is 5. The gap

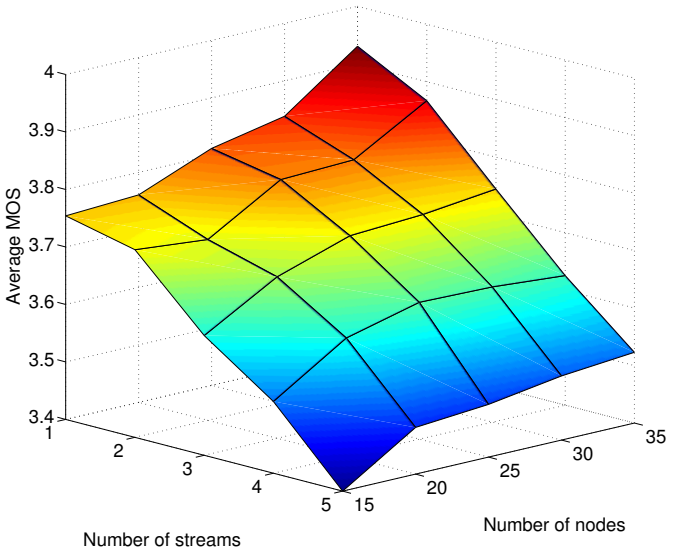


Figure 9: Average MOS under variations of the number of nodes (N) and the number of streams (D)

between OLSR and the proposed scheme can be up to 1.0 in MOS. The gap between 10-iteration AD^3-GLAM and 100-iteration AD^3-GLAM is less than 0.1 in terms of MOS, which is not detectable by humans. Meanwhile, there is no difference in performance when the number of iterations increases from 100 to 1000.

In Fig. 11, the number of nodes is 25. The gap between OLSR and AD^3-GLAM is significant when the number of streams is high. When there is only one stream in the network, the interference impact is ignorable. Therefore, forwarding packets through the shortest path does have similar performance to AD^3-GLAM. However, the interference problem becomes more severe when the number of streams increases. Consequently, AD^3-GLAM, which solves the optimization problem considering interference, will significantly outperform OLSR. The gap between AD^3-GLAM and OLSR can be up to 0.7 in MOS.

Next, we evaluate the performance of AD^3-GLAM by comparing it with the exact solution. The exact solution is obtained by using well-known solver called Gurobi [52].

We conduct the simulations with three different number of nodes: 15, 20, and 25. Meanwhile, the number of streams is still from 1 to 5. The approximation ratio (AR) is defined as the ratio of the objective value of AD^3-GLAM with 100 iterations and the exact solution. Fig. 12 shows the AR under different simulation configurations. The AR values in all configurations are over 90% and decrease when the number of nodes or the number of streams increases. In other words, the AR of AD^3-GLAM decreases as the problem size increases. Besides

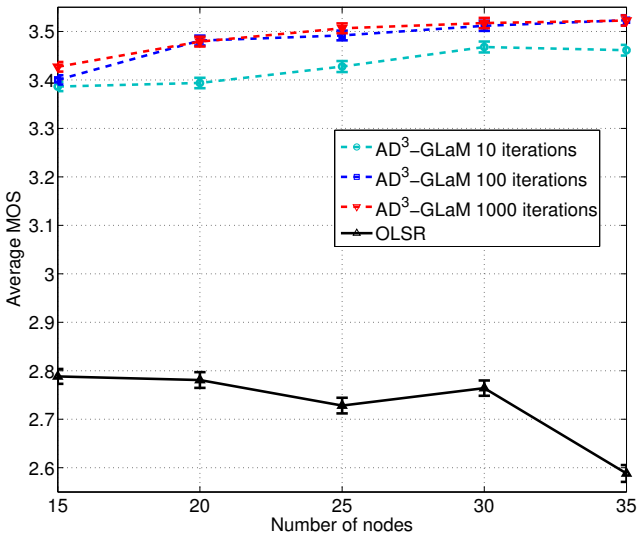


Figure 10: Performance of AD^3-GLaM and OLSR under variations of the number of nodes (N)

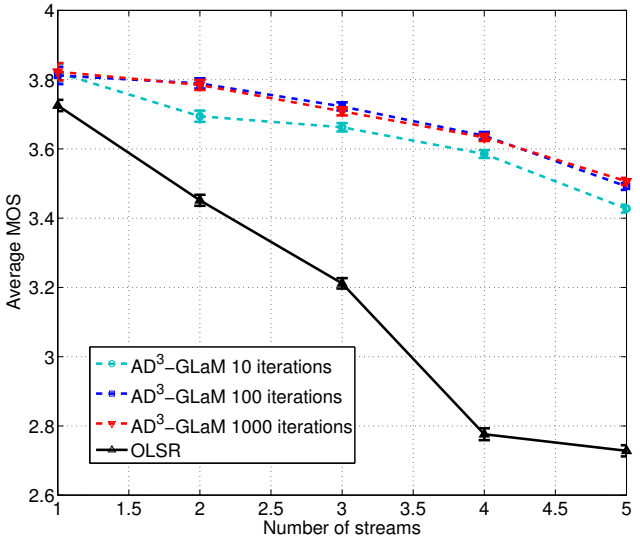


Figure 11: Performance of AD^3-GLaM and OLSR under variations of the number of streams (D)

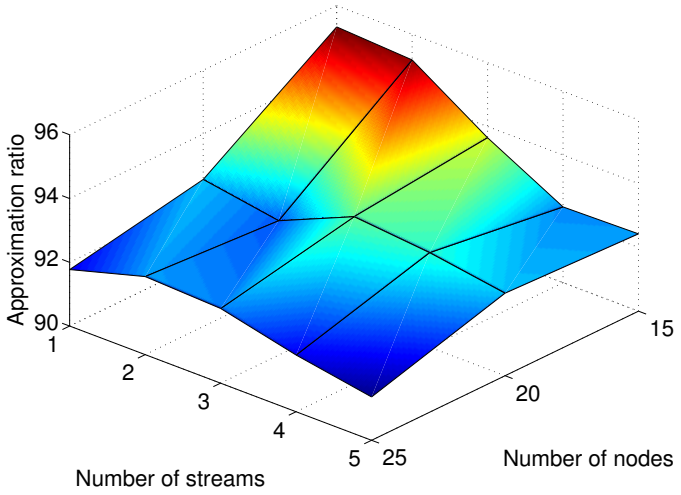


Figure 12: Approximation ratio

AR, the absolute gap between two approaches is also meaningful. Fig. 13 demonstrates the gap between the exact and AD^3-GLAM solutions. Although the AR is high, the gap in MOS between the two approaches is not negligible. It represents the trade-off for running a decentralized algorithm. The calculation time of Gurobi, nevertheless, is much higher than that of AD^3-GLAM with 100 iterations as shown in Fig. 14. For instance, the calculation time of Gurobi of 25-node case is 4000 times as much as the calculation time of AD^3-GLAM.

7.2.2. Fairness

Besides average MOS, the fairness is also important aspect that should be taken into account. To measure the fairness of AD^3-GLAM, we adopt Jain's fairness index which is determined as follows.

$$\mathcal{J}\left(\Psi_1, \Psi_2, ..., \Psi_n\right) = \frac{\left(\sum_{i=1}^{n} \Psi_i\right)^2}{n \times \sum_{i=1}^{n} \Psi_i^2}, \tag{29}$$

where Ψ_i is the MOS of stream i. Fig. 15 shows the fairness of OLSR and AD^3-GLAM when the number of streams is 5. AD^3-GLAM provides better fairness indexes as compared to OLSR with the gap of about 0.08. There is an insignificant impact on the fairness when varying the number of iterations of AD^3-GLAM.

All above simulations confirm the performance of AD^3-GLAM. Now, we analyze the cost (in calculation time) of AD^3-GLAM under various numbers of iterations. Fig. 16 demonstrates the calculation time with different numbers

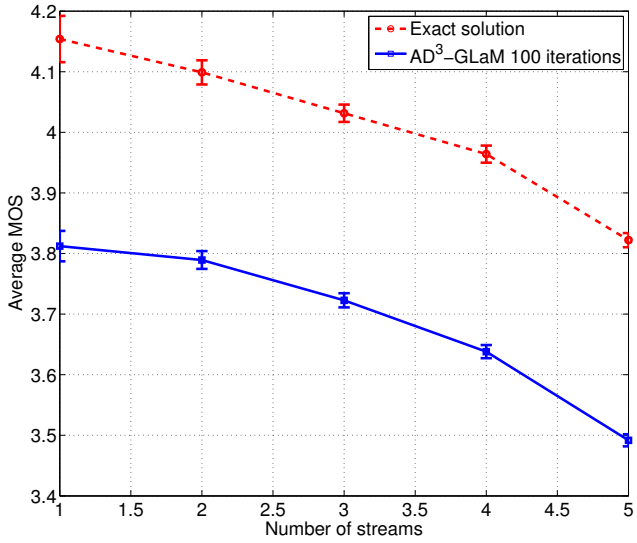


Figure 13: Exact solution vs AD^3-GLAM 100 iteration solution

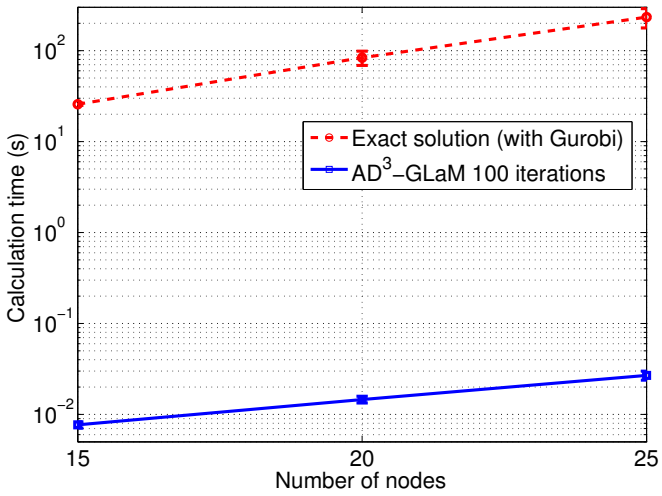


Figure 14: Calculation time of AD^3-GLAM and Gurobi

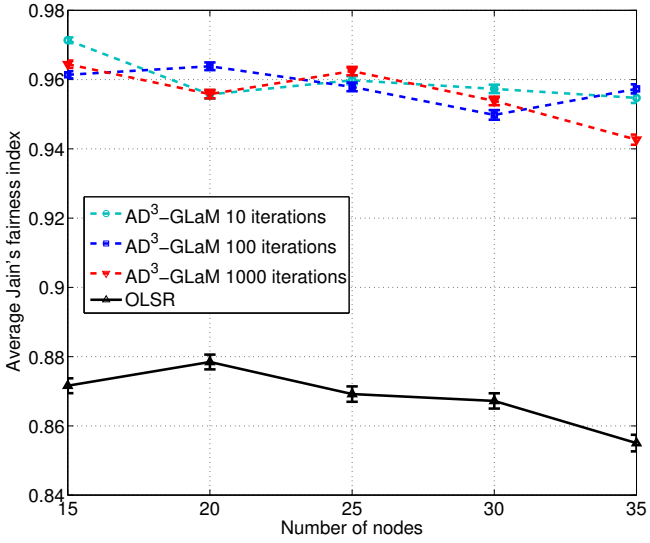


Figure 15: Fairness

of iterations. While the calculation time values of AD^3-GLAM 10 iterations and AD^3-GLAM 100 iterations do not differ much, the calculation time with AD^3-GLAM 1000 iterations is about 5 times greater than that of AD^3-GLAM 10 iterations and AD^3-GLAM 100 iterations. The complexity of the problem is in proportion to the size of the networks, so the calculation time increases with the number of nodes. Additionally, a higher number of iterations also mean higher computation time. Meanwhile, the average MOS is slightly enhanced when increasing the number of iterations as shown in Fig. 11. Consequently, to find a compromise, it is favorable to select a low number of iterations in small and moderate network size, above which the gains in average MOS become marginal.

7.3. Overhead

The total number of messages per iteration, obtained by solving the optimization Problem 2 (Opt. Msg.) and in worst case (WC Msg.), are compared in this section. The worst case consists in factors such that all variables related to them are assigned to different GW. Fig. 17 shows the total number of messages per iteration in two cases when the number of streams is 5. The assignment of optimization Problem 2 can help to reduce up to 3000 messages. Assume that the length of each message is 64-bit so as to convey a floating point value, the gain in overhead size per iteration is $3000 \times 64 = 192000$ bits. By optimally allocating factors and variables to GWs, 'Opt. Msg.' is able to significantly reduce the number of messages exchanged between GWs during computation.

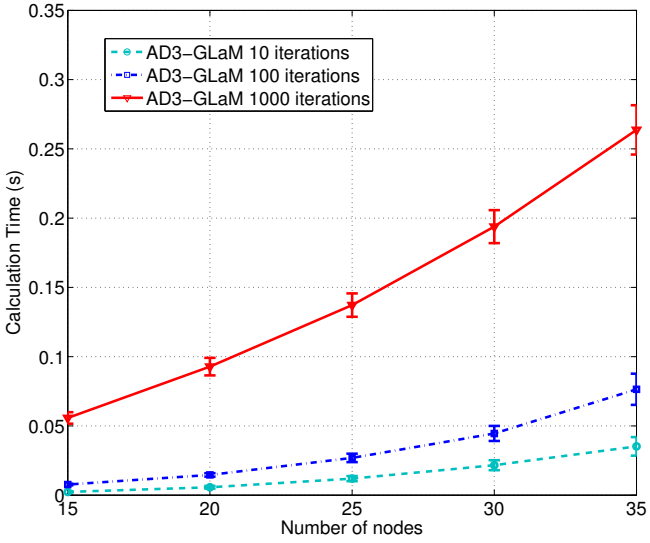


Figure 16: Calculation time

The number of messages under different numbers of streams and numbers of nodes is shown in Fig. 18.

8. Conclusion

In this paper, we studied a QoE-based joint routing and layer-allocation problem for SVC video streaming in WMNs. This is the first research on adopting AD^3, an advanced message passing algorithm, to solve a routing problem in wireless networking. We combined AD^3 and a distributed heuristic decoding algorithm so as to determine the sub-optimal solution. AD^3-GLaM runs on OLSR which is a conventional routing protocol and is available on most of modern ad-hoc supporting devices. That makes it easy to implement AD^3-GLaM in practical world.

Additionally, we considered the factor and variable assignment problem to address the high cost of exchanging messages in networks. The optimal solution derived by solving the optimization problem can reduce significantly the total number of exchanging messages.

The intensive simulation results confirmed that AD^3-GLaM is not only better than conventional OLSR, but is also asymptotic to exact optimal solutions. Moreover, AD^3-GLaM also provides a good fairness among users which in turn is also an important aspect in video streaming.

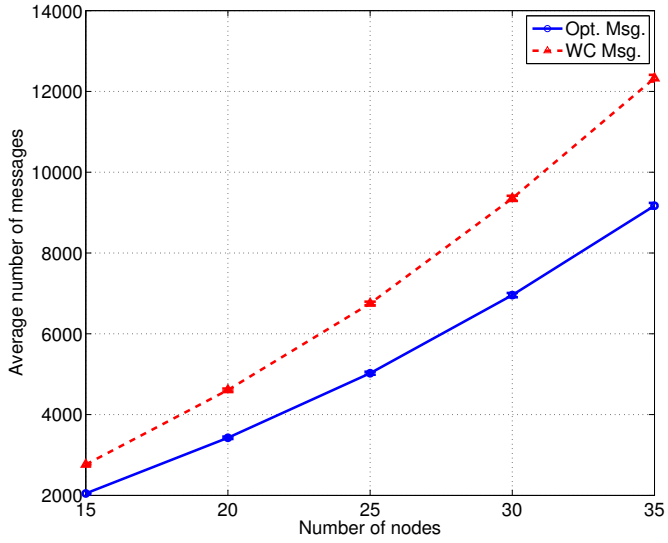


Figure 17: The total number of messages per iteration in worst and optimal cases

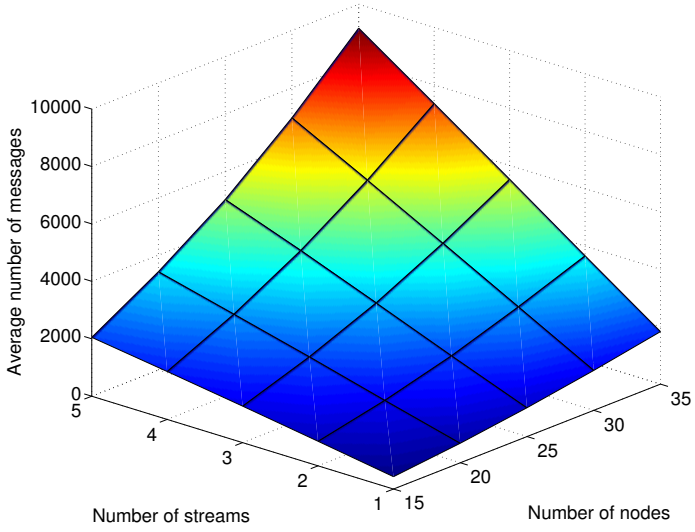


Figure 18: The optimal number of messages under different numbers of nodes and numbers of streams

Acknowledgements

This paper has been done in the context of a scientific collaboration between IRISA, Laboratoire Hubert Curien, CReSTIC and IIIA-CSIC. This work has been partly supported by EIT Digital and projects Collectiveware TIN2015-66863-C2-1-R (MINECO/FEDER) and 2014 SGR 118.

[1] J. McDermott, A majority of us mobile users are now smart phone users, Ad Age. May 28.

[2] H. Luo, R. Ramjee, P. Sinha, L. E. Li, S. Lu, UCAN: A Unified Cellular and Ad-hoc Network Architecture, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, MobiCom '03, ACM, New York, NY, USA, 2003, pp. 353–367. doi:10.1145/938985.939021.

[3] P. K. McKinley, H. Xu, A. H. Esfahanian, L. M. Ni, Unicast-based multicast communication in wormhole-routed networks, IEEE Transactions on Parallel and Distributed Systems 5 (12) (1994) 1252–1265. doi:10.1109/71.334899.

[4] H. Wu, C. Qiao, S. De, O. Tonguz, Integrated cellular and ad hoc relaying systems: iCAR, IEEE Journal on Selected Areas in Communications 19 (10) (2001) 2105–2115. doi:10.1109/49.957326.

[5] J. Zhou, Y. Yang, PAR CelS: Pervasive ad-hoc relaying for cell systems, in: Proc. IFIP Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 2002.

[6] Cisco visual networking index: Forecast and methodology, 2015-2020, Tech. rep., Cisco (2016).

[7] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard, IEEE Trans. Circuits Syst. Video Technol. 17 (9) (2007) 1103–1120. doi:10.1109/TCSVT.2007.905532.

[8] C. Muller, D. Renzi, S. Lederer, S. Battista, C. Timmerer, Using scalable video coding for dynamic adaptive streaming over http in mobile environments, in: EUSIPCO, 2012, pp. 2208–2212.

[9] Methodology for the subjective assessment of the quality of television pictures (Mar. 2000).

[10] S. Mohamed, G. Rubino, A study of real-time packet video quality using random neural networks, IEEE Trans. Circuits Syst. Video Technol. 12 (12) (2002) 1071–1083. doi:10.1109/TCSVT.2002.806808.

[11] K. Deep Singh, K. Piamrat, H. Park, C. Viho, J.-M. Bonnin, Optimising QoE for Scalable Video multicast over WLAN, in: IEEE PIMRC, 2013, pp. 2131–2136. doi:10.1109/PIMRC.2013.6666496.

[12] Y. Li, L. Zhou, Y. Yang, H.-C. Chao, Optimization architecture for joint multi-path routing and scheduling in wireless mesh networks, Mathematical and Computer Modelling 53 (3–4) (2011) 458 – 470.

[13] P. Cappanera, L. Lenzini, A. Lori, G. Stea, G. Vaglini, Optimal joint routing and link scheduling for real-time traffic in TDMA Wireless Mesh Networks, Computer Networks 57 (11) (2013) 2301 – 2312.

[14] H. Nam, K. H. Kim, J. Y. Kim, H. Schulzrinne, Towards QoE-aware video streaming using SDN, in: 2014 IEEE Global Communications Conference, 2014, pp. 1317–1322. doi:10.1109/GLOCOM.2014.7036990.

[15] P. T. A. Quang, K. Piamrat, K. D. Singh, C. Viho, Qoe-based routing algorithms for h.264/svc video over ad-hoc networks, Wireless Networks 22 (7) (2016) 2387–2402. doi:10.1007/s11276-015-1103-0.

[16] P. T. A. Quang, K. Piamrat, K. D. Singh, C. Viho, Video streaming over ad hoc networks: A qoe-based optimal routing solution, IEEE Transactions on Vehicular Technology 66 (2) (2017) 1533–1546. doi:10.1109/TVT.2016.2552041.

[17] X. Fang, D. Yang, G. Xue, Consort: Node-Constrained Opportunistic Routing in wireless mesh networks, in: IEEE INFOCOM, 2011, pp. 1907–1915. doi:10.1109/INFCOM.2011.5934993.

[18] M. Xiao, J. Wu, C. Liu, L. Huang, TOUR: Time-sensitive Opportunistic Utility-based Routing in delay tolerant networks, in: IEEE INFOCOM, 2013, pp. 2085–2091. doi:10.1109/INFCOM.2013.6567010.

[19] N. Kumar, J. H. Lee, J. J. P. C. Rodrigues, Intelligent Mobile Video Surveillance System as a Bayesian Coalition Game in Vehicular Sensor Networks: Learning Automata Approach, IEEE Transactions on Intelligent Transportation Systems 16 (3) (2015) 1148–1161. doi:10.1109/TITS.2014.2354372.

[20] A. F. Martins, F. Mário AT, A. Pedro MQ, N. A. Smith, E. P. Xing, AD^3: Alternating directions dual decomposition for map inference in graphical models, Journal of Machine Learning Research 16 (2015) 495–545.

[21] M. Wang, Y. Cui, X. Wang, S. Xiao, J. Jiang, Machine learning for networking: Workflow, advances and opportunities, IEEE Network 32 (2) (2018) 92–99. doi:10.1109/MNET.2017.1700200.

[22] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, Tech. rep., IETF (2003).

[23] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, in: IEEE INMIC, 2001, pp. 62–68. doi:10.1109/INMIC.2001.995315.

[24] A. Huhtonen, Comparing AODV and OLSR routing protocols, Telecommunications Software and Multimedia (2004) 1–9.

[25] J. Yi, A. Adnane, S. David, B. Parrein, Multipath optimized link state routing for mobile ad hoc networks, Ad Hoc Networks 9 (1) (2011) 28 – 47.

[26] Y. Peng, Y. Yu, L. Guo, D. Jiang, Q. Gai, An efficient joint channel assignment and QoS routing protocol for IEEE 802.11 multi-radio multi-channel wireless mesh networks, Journal of Network and Computer Applications 36 (2) (2013) 843 – 857. doi:http://dx.doi.org/10.1016/j.jnca.2012.11.003.

[27] U. Lee, S. F. Midkiff, OLSR-MC: A proactive routing protocol for multichannel wireless ad-hoc networks, in: IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006., Vol. 1, 2006, pp. 331–336. doi:10.1109/WCNC.2006.1683486.

[28] H. Xie, A. Boukerche, A. A. F. Loureiro, A multipath video streaming solution for vehicular networks with link disjoint and node-disjoint, IEEE Transactions on Parallel and Distributed Systems 26 (12) (2015) 3223–3235. doi:10.1109/TPDS.2014.2371027.

[29] H. A. Tran, A. Mellouk, S. Hoceini, S. Souihi, User QoE-based adaptive routing system for future Internet CDN, in: Computing, Communications and Applications Conference (ComComAp), 2012, 2012, pp. 90–95. doi:10.1109/ComComAp.2012.6154009.

[30] P. A. K. Acharya, D. L. Johnson, E. M. Belding, Gateway-aware routing for wireless mesh networks, in: The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010), 2010, pp. 564–569. doi:10.1109/MASS.2010.5663893.

[31] A. Zhou, M. Liu, Z. Li, E. Dutkiewicz, Cross-layer design with optimal dynamic gateway selection for wireless mesh networks, Computer Communications 55 (2015) 69 – 79. doi:http://dx.doi.org/10.1016/j.comcom.2014.08.011.

[32] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122. doi:10.1561/2200000016.

[33] M. J. Wainwright, T. S. Jaakkola, A. S. Willsky, Tree-reweighted belief propagation algorithms and approximate ml estimation by pseudo-moment matching, in: Workshop on Artificial Intelligence and Statistics, Vol. 21, Society for Artificial Intelligence and Statistics Np, 2003, p. 97.

[34] N. Komodakis, N. Paragios, G. Tziritas, Mrf optimization via dual decomposition: Message-passing revisited, in: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, 2007, pp. 1–8.

[35] A. Globerson, T. S. Jaakkola, Fixing max-product: Convergent message passing algorithms for map lp-relaxations, in: Advances in neural information processing systems, 2008, pp. 553–560.

[36] T. Hazan, A. Shashua, Norm-Product Belief Propagation: Primal-Dual Message-Passing for Approximate Inference, IEEE Transactions on Information Theory 56 (12) (2010) 6294–6316. doi:10.1109/TIT.2010.2079014.

[37] F. Cruz-Mencia, J. Cerquides, A. Espinosa, J. C. Moure, J. A. Rodriguez-Aguilar, Parallelisation and Application of AD^3 as a Method for Solving Large Scale Combinatorial Auctions, in: International Conference on Coordination Languages and Models, Springer International Publishing, 2015, pp. 153–168.

[38] C. Yanover, T. Meltzer, Y. Weiss, Linear programming relaxations and belief propagation–an empirical study, Journal of Machine Learning Research 7 (Sep) (2006) 1887–1907.

[39] P. T. A. Quang, K. Piamrat, C. Viho, QoE-Aware Routing for Video Streaming over VANETs, in: IEEE 80th Vehicular Technology Conference (VTC Fall), 2014, pp. 1–5. doi:10.1109/VTCFall.2014.6966141.

[40] P. T. A. Quang, K. Piamrat, C. Viho, QoE-aware routing for video streaming over ad-hoc networks, in: 2014 IEEE Global Communications Conference, 2014, pp. 181–186. doi:10.1109/GLOCOM.2014.7036804.

[41] T. Penya-Alba, M. Vinyals, J. Cerquides, J. A. Rodriguez-Aguilar, Exploiting max-sum for the decentralized assembly of high-valued supply chains, in: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS '14, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2014, pp. 373–380.
URL `http://dl.acm.org/citation.cfm?id=2615731.2615793`

[42] E. Perahia, A. Sheth, T. Kenney, R. Stacey, D. Halperin, Investigation into the doppler component of the ieee 802.11n channel model, in: IEEE Globecom, 2010, pp. 1–5. doi:10.1109/GLOCOM.2010.5684207.

[43] A. Woo, T. Tong, D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03, ACM, New York, NY, USA, 2003, pp. 14–27. doi:10.1145/958491.958494.

[44] X. Deng, L. He, Q. Liu, X. Li, L. Cai, Z. Chen, EPTR: expected path throughput based routing protocol for wireless mesh network, Wireless Networks 22 (3) (2016) 839–854. doi:10.1007/s11276-015-1003-3.

[45] L. Xiao, M. Johansson, S. Boyd, Simultaneous routing and resource allocation via dual decomposition, IEEE Trans. Commun. 52 (7) (2004) 1136–1144. doi:10.1109/TCOMM.2004.831346.

[46] Y. Yi, S. Shakkottai, Hop-by-hop congestion control over a wireless multi-hop network, IEEE/ACM Trans. Netw. 15 (1) (2007) 133–144. doi:10.1109/TNET.2006.890121.

[47] N. M. Do, C.-H. Hsu, N. Venkatasubramanian, Video dissemination over hybrid cellular and ad hoc networks, IEEE Trans. Mobile Comput. 13 (2) (2014) 274–286. doi:10.1109/TMC.2012.246.

[48] B. Hajek, G. Sasaki, Link scheduling in polynomial time, IEEE Trans. Inf. Theory 34 (5) (1988) 910–917. doi:10.1109/18.21215.

[49] T. Vanhatupa, Wi-fi capacity analysis for 802.11ac and 802.11n: Theory & practice, Tech. rep., Ekahau Inc. (2013).

[50] R. Cohen, L. Katzir, The generalized maximum coverage problem, Information Processing Letters 108 (1) (2008) 15 – 22. doi:http://dx.doi.org/10.1016/j.ipl.2008.03.017.

[51] M. B. Almeida, A. F. Martins, Fast and robust compressive summarization with dual decomposition and multi-task learning., in: ACL (1), 2013, pp. 196–206.

[52] Gurobi, https://www.gurobi.com/.

[53] Cplex, https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[54] Ns-3, https://www.nsnam.org/.